

第一章

一. 名词解释

1. 是负责管理计算机软硬件的一组软件集合
4. 能够实时处理外部事件的操作系统, 在严格时限内响应外部事件
5. 资源一次仅允许一个进程使用, 其他进程需等待

二.

2. A2 B2 C. (4)

3. A. 4 B. 3 C. 4

三. (1) 硬件系统 软件系统

(3) 硬件资源 软件资源

(4) 共享性 异步性

第二章

3. 访问共享资源的代码段, 需互斥执行

4. 协调并发进程执行顺序, 使多个进程按一定规则协同工作的机制

二. 1. A2 B2 C2 D2 3. A2 B2 C2 D2

三. 1.

5. 动态 静态

6. 间接制约

第

页

8. 共享内存 消息传递 管道通信

9. 就绪 执行 阻塞

五

2. $S_2 \rightarrow S_1, S_3 \rightarrow S_1, S_4 \rightarrow S_2, S_5 \rightarrow S_2, S_6 \rightarrow S_3, S_7 \rightarrow S_4, S_8$

| | | | | |
|-------------|------------|------------|------------|----------|
| semaphore a | $\{P(a)\}$ | $\{P(a)\}$ | $\{P(c)\}$ | S_8 |
| semaphore b | S_1 | $P(a)$ | $P(b)$ | |
| semaphore c | $V(a)$ | S_2 | S_3 | |
| semaphore d | $V(b)$ | $V(a)$ | $V(c)$ | |
| semaphore e | $P(c)\}$ | $P(b)\}$ | $P(b)\}$ | $P(f)\}$ |
| semaphore f | $P(c)$ | $P(a)$ | $P(e)$ | $P(g)$ |
| semaphore g | $V(f)$ | S_5 | S_6 | $P(h)$ |
| semaphore h | $\}$ | $\}$ | $\}$ | S_7 |

3. semaphore plate = 5
 semaphore apple = 0
 semaphore orange = 0
 semaphore mutex = 1

| | |
|------------------|----------------|
| father { | son { |
| while (true) { | while (true) { |
| P(plate) | P(plate) |
| P(mutex) | P(mutex) |
| 放入水果; | 吃橘子 |
| if (苹果) V(apple) | V(plate) |
| else V(orange) | V(mutex) |
| V(mutex) | } |
| } | |

daughter {
 while (true) {
 P(apple)
 P(mutex)
 吃苹果
 V(plate)
 V(mutex)
 }
 }

4. semaphore empty1 = 1 full1 = 0

semaphore empty2 = 1 full2 = 0

PAc > {

PBc > {

PCc > {

while (true) {

while (true) {

while (true) {

p(empty1);

p(full1);

p(full2);

读入缓冲1;

p(empty2);

打印;

v(full1);

复制;

v(empty2);

}

v(empty1);

}

v(full2);

}

5. ~~semaphore nobody = 0~~

semaphore waiting = 0

semaphore action = 1

semaphore consumer = 0;

hairer {

customer {

while (true) {

~~p(nobody);~~

~~p(action);~~

p(consumer);

if (waiting < N) {

付款;

waiting--;

waiting++;

else 离开;

理发;

v(customer);

收费;

p(action);

}

理发;

第三章

- 一. 2. 从就绪队列选择进程分配CPU
3. 作业从提交到完成的时间间隔
4. 多个进程因资源竞争而无限等待

- 二. 2. A:2 B:3 C:3 D:4 E:4 F:2
3. A:5 B:2 C:3 D:3 E:2
4. A:2 B:1

- 三. 2. 提交 后备 完成

6. 预防死锁 避免死锁 检测死锁 检测死锁

| | 完成时间 | 周转时间 | 带权周转时间 | 平均周转时间 |
|---------------|------|------|--------|-------------------|
| 五. 1. FCFS: A | 3 | 3 | 1 | |
| B | 9 | 7 | 1.17 | |
| 平均带权周转: 2.6 | C | 13 | 2.25 | |
| 平均周转: 8.6 | D | 18 | 2.4 | |
| E | 20 | 12 | 6 | |
| 2. 非抢占: A | 3 | 3 | 1 | |
| B | 9 | 7 | 1.17 | |
| 平均周转: 7.6 | C | 15 | 2.75 | |
| 平均带权周转: 1.844 | D | 20 | 2.8 | |
| E | 11 | 3 | 1.5 | |
| 3. 抢占: A | 3 | 3 | 1 | |
| B | 15 | 13 | 2.17 | |
| 平均周转: 7.2 | C | 8 | 2 | |
| 平均带权周转: 1.6 | D | 20 | 2.8 | |
| E | 10 | 14 | 1 | |
| 4. RR | A | 4 | 1.3 | |
| 平均周转: 10.8 | B | 18 | 2.7 | |
| 平均带权周转: 2.7 | C | 17 | 3.3 | |
| E | 20 | 14 | 2.8 | |
| | 15 | 7 | 3.5 | |
| 5. HRRN | A | 3 | 1 | |
| B | 9 | 7 | 1.17 | |
| 平均周转: 8 | C | 13 | 2.25 | |
| 平均带权周转: 2.14 | D | 20 | 2.8 | |
| E | 15 | 7 | 3.5 | |

2. 每次选择松弛度最小的进程进行调度

当 $s = 0ms$ 时, $\alpha_1 = 0.5$ $\alpha_2 = 0.2$ $\alpha_3 = 0.3$ 选 P_1

选 P_1

3. 4矩阵:

| | | |
|---|---|----|
| 0 | 0 | 12 |
| 1 | 7 | 5 |
| 2 | 3 | 5 |
| 0 | 6 | 5 |
| 0 | 6 | 5 |

设 P_1 已分配矩阵值

~~无法计算~~

是安全的, 能找到安全序列

$P_0 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1 \rightarrow P_2$

② 不能分配, 分配以后可用为 $(0, 4, 0, 0)$

无法满足任何一个进程需求, 不安全

③ 不会立即死锁